Journal of Computational and Applied Mathematics 18 (1987) 307-320
North-Holland

A parallel subgradient projections method for the convex feasibility problem

Lucio Tunes DOS SANTOS
ICMSC- USP-São Carlos, 13 *560-São Carlos-SP, Brazil*

Received 20 December 1985
Revised 20 March 1986

Abstract: An iterative method to solve the convex feasibility problem for a finite family of convex sets is presented. The algorithm consists in the application of a generalization of an acceleration procedure introduced by De Pierro, in a parallel version of the Subgradient Projections Method proposed by Censor and Lent. The generated sequence is shown to converge for any starting point. Some numerical results are presented.

Keywords: Convex programming, iterative methods, non-linear inequalities, projection methods.

1. Introduction

Let Q be a closed convex subset of $\mathbb{R}^n$ and let g,: $\mathbb{R}^n \to \mathbb{R}$, $i = 1,\ldots,m$ be convex functions. It is considered the computation of a solution x * for the system of convex inequalities:

$$x \in Q,\ g_i(x) \leqslant 0,\ i = 1,\ldots,\ m\ .$$

This problem is known as the Convex Feasibility Problem and arises in different fields, for example, in the series expansion approach to image reconstruction from projections (see [2]). Let S be denoted the solution set, i.e.,

$$S = \{x \in Q / g_i(x) \leqslant 0, i = 1,\ldots,\ m\}.$$

Gubin et al. [8] have proposed a generalization of Kaczmarz method [9], called the Successive Orthogonal Projections method, to solve this problem.

Kaczmarz's method and its generalizations, called by Censor [1] Row-Action Methods, make no changes in the original system, perform no operation on the system as a whole, and require access to only one component at a time. These are the reasons why the storage requirements for these methods are very low, in comparison with traditional methods.

For the linear case, Censor and Elfving [3] have developed an algorithm, which is an extension of the Cimmino's method [5] for linear equalities, where each iterate is reflected on all the hyperplanes and the new iterate is on the line determined by the previous iterate and a convex combination of the projections. De Pierro and Iusem [7] have extended these algorithms along the lines of Gubin et al., i.e., by considering general convex sets in a Hilbert space instead of hyperplanes or half-spaces.

Censor and Lent [4] have proposed a method where the orthogonal projection in the convex sets is replaced by the projection of the (sub)gradient. This method eliminates the solution, at each step, of a *subsidiary minimization problem* associated with the projection onto the current set.

On the other hand, some authors have incorporated acceleration procedures in order to improve the speed of convergence of different row-actions methods (see for example [10,14]). In this paper a version of the Censor and Lent's method is accelerated based on the lines of De Pierro and Iusem, i.e., by taking convex combinations of the approximated projections and using a generalization of the scheme introduced by De Pierro [6] for linear systems. The idea of the acceleration is simple: given two consecutives iterates x^K and x^{K+1}, the accelerated iterate is an approximation to the point on the line $\{x^K, x^{K+1}\}$, which is closest to the solution.

In Section 2 of this paper some introductory results are given. In Section 3 the Censor and Lent's method using parallel projections, i.e., convex combinations of the (sub)gradient projections, is presented. The De Pierro's scheme of acceleration for parallel projections in the linear case is applied in Section 4. The new method and convergence theorem are presented in Section 5. Section 6 presents some numerical experiences.

Finally, in Section 7 some conclusions are stated and the lines for future research are suggested.

2. Preliminaries

In this section some notations are presented and some basic results, which will be used later, are given.

2.1. *Orthogonal projection*

Given a set Q $c\ \mathbb{R}^n$ and a point x $\in \mathbb{R}^n$, $P_Q(x)$ will denote the orthogonal projection of x onto Q, i.e., a point for which

$$\| x - P_Q(x) \| = \inf_{y \in Q} \| x - y \|$$

where $\|\cdot\|$ denotes the Euclidian norm in $\mathbb{R}^n$.

If Q is non-empty and closed then P,(x) always exists. In addition, if Q is also convex then $P_Q(x)$ is uniquely determined by x. The projection operator is non-expansive, i.e..

$$\| P_Q(x) - P_Q(y) \| \leqslant \| x - y \| \quad \text{for every } x, y \in \mathbb{R}^n.$$

2.2. *Subgradients*

A vector $t \in \mathbb{R}^n$ is said to be a subgradient of a convex function g at a point y if (t, x-y} <g(x)-g(y) for every $x \in \mathbb{R}^n$ ($\langle \cdot, \cdot \rangle$ denote the inner product in $\mathbb{R}^n$). This inequality is referred to as the subgradient inequality. If g is differentiable at y then its gradient $\nabla g(y)$ is the unique subgradient of g at y. A convex function always has a subgradient. The set of all subgradients of g at y is denoted by $\partial g(y)$.

2.3. *Kinks*

The function g'(x) obtained from a given real valued function g(x), defined on $\mathbb{R}^n$ via the operator g'(x) = max{ 0, g(x)}, is called the kink of g. If g(x) is convex then $g^+(x)$ is convex and min g'(x) = 0. It is always true that $\{x \in \mathbb{R}^n / g(x) \leqslant 0\} = \{x \in \mathbb{R}^n / g^+(x) = 0\}$.

3. **A parallel subgradient projections method**

The Censor and Lent's method, called *cyclic subgradient projections method* (CSP), is an iterative method which uses the convex sets one at a time in each iterative step, where is required (sub)gradient calculation of the particular convex function taken up and does not rely on orthogonal projections onto the convex sets described by the convex inequalities. The idea is as follows:

Given the iterate x^k, the CSP method makes a move from x^k in the direction of the negative (sub)gradient of g_{i_k} (the function taken up in this step) at the point x^k *itself.* Such a move might be 'less efficient' than an orthogonal projection in terms of rate of convergence of the whole process, but is enables us to replace the subsidiary minimization problems by (sub)gradient calculations (see Fig. 1). The functions are taken up in a cyclic way.

In a first adaptation, in each iteration is applied the CSP method for all the convex sets and the new iterative lies on the half line defined by the previous one and a convex combination (with fixed coefficients) of these approximated projections. The location on the line being determined by a relaxation parameter. This method was called Parallel Subgradient Projections Method (PSP) (for more details see Santos [13]). The typical step is:

$$x^{k+1} = x^k - \alpha_k \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)}{\|t_i^k\|^2} t_i^k$$

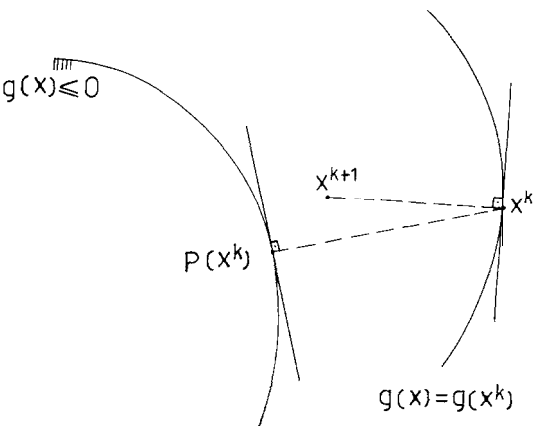

Fig. 1. The CSP method.

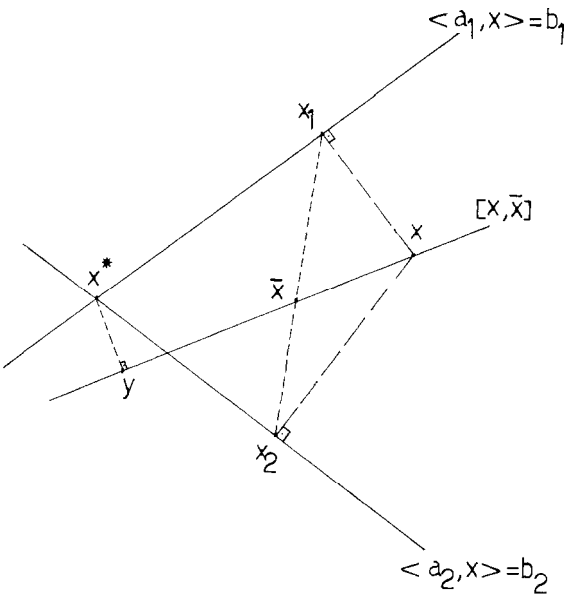

Fig. 2. The acceleration procedure.

where $t_i^k \in \partial g_i^+(x^k)$, $\{\alpha_k\}_{k=0}^{\infty}$ is a sequence of relaxation parameters confined into the interval (0, 2) and $0 < \lambda_i < 1$, $i = 1, \ldots, m$ with $\sum_{i=1}^m \lambda_i = 1$.

Note that if it is possible the use of parallel computation (see [12]) a iteration of the PSP method is equivalent, in computer time, to a projection in the CSP method.

4. **Acceleration procedure**

Here the scheme of acceleration, introduced by De Pierro, is applied, to the Cimmino's method. Let x be the iterate in some step and the projections of x onto all the hyperplanes $(a_i, x) = b_i$, $i = 1, \ldots, m$, be calculated:

$$x_i = x - u_i a_i, \quad i = 1, \ldots, m$$

with

$$u_i = (\langle a_i, x \rangle - b_i) / \| a_i \|^2.$$

Let $\bar{x}$ be a convex combination of the x_i's, $\bar{x} = \sum_{i=1}^m \lambda_i x_i$ with $0 < \lambda_i < 1$ and $\sum_{i=1}^m \lambda_i = 1$.

The accelerate iterate y, is the point on the line $[x, \bar{x}]$ closest to the solution x^* (see Fig. 2, for the case $m = 2$), i.e.,

$$y = x + \gamma(\bar{x} - x), \quad \gamma \in \mathbb{R}$$

and

$$\langle x^* - y, \bar{x} - x \rangle = 0$$

(we suppose nonsingularity).

Then,

$$\langle x^* - y,\ \text{X-x}\rangle = \langle x^* - \text{x} - \text{y}(\text{x} - \text{x}),\quad \text{X} - \text{x}\rangle$$
$$= \langle x^* - \text{x},\ \bar{x} - x\rangle \gamma \| \bar{x} - x \|^2 = 0.$$

So,

$$\gamma = \langle x^* - \text{x},\ \bar{x} - x\rangle / \| \bar{x} - \text{x} \|^2.$$

Now,

$$\bar{x} - \text{x} = \sum_{i=1}^{m} \lambda_i x_i - \text{x} = \sum_{i=1}^{m} \lambda_i (x_i - \text{x}) = \sum_{i=1}^{m} \lambda_i (-u_i a_i)$$
$$= \sum_{i=1}^{m} \lambda_i u_i a_i = -\sum_{i=1}^{m} w_i a_i = -A^{\mathrm{T}} W$$

where

$$A^{\mathrm{T}} = [a, \ldots \sim,] \quad \text{and} \quad W = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \quad \text{with } w_i = \lambda_i u_i.$$

Therefore,

$$\langle x^* - x,\ \text{X-x}\rangle = (\text{x}^* - \text{x},\ -A^{\mathrm{T}} W\rangle = \langle Ax - Ax^*,\ W)$$
$$= \langle Ax - b, W\rangle$$

where

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

This implies that

$$y = x + \frac{\langle Ax - b, W\rangle}{\| \bar{x} - x \|^2} (\bar{x} - x)$$

and introducing a relaxation parameter α, confined into the interval (0, 2), we have

$$\bar{y} = x + \alpha (y - x)$$

or

$$\bar{y} = x + \alpha \frac{\langle Ax - b, W\rangle}{\| \bar{x} - x \|^2} (\bar{x} - x).$$

Now, the expression of $\bar{y}$ for the nonlinear case applying the PSP is generalized. It is known that g(x) $\leqslant$ 0 is equal to g'(x) = 0. Then,

$$\gamma = \langle G^+(x), W\rangle / \| \bar{x} - x \|^2$$

where

$$G^+(x) = \begin{bmatrix} g_1^+(x) \\ \vdots \\ g_m^+(x) \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

with

$$w_i = g_i^+(x)/\| t_i \|^2, \quad t_i \in \partial g_i^+(x).$$

Denoting

$$v = x - \bar{x} = x - \sum_{i=1}^{m} \lambda_i x_i = \sum_{i=1} \lambda_i u_i t_i = \sum_{i=1} \lambda_i \frac{g_i^+(x)}{\| t_i \|^2} t_i,$$

$$\beta = \langle G^+(x), W \rangle = \sum_{i=1}^{m} g_i^+(x) w_i = \sum_{i=1}^{m} \lambda_i \frac{(g_i(x))^2}{\| t_i \|^2},$$

$$\bar{y} = x - \alpha \cdot \beta \cdot v/\| v \|^2 \quad \text{with } \alpha \in (0, 2).$$

5. The PSP method with acceleration

At the present moment the basic ideas have been introduced. Therefore the new method (abbreviated by PSPA) is presented:

Algorithm

Initialization: $x^0 \in Q$ is arbitrary.

Typical step: $x^{k+1} = P_Q(\tilde{x}^{k+1})$, where

$$\tilde{x}^{k+1} = x^k - \alpha_k \beta_k v^k/\| v^k \|^2,$$

$$\beta_k = \sum_{i=1}^{m} \lambda_i \left(g_i^+(x^k)\right)^2/\left\| t_i^k \right\|^2,$$

$$v^k = \sum_{i=1}^{m} \lambda_i g_i^+(x^k)/\left\| t_i^k \right\|^2 t_i^k$$

where $t_i^k \in \partial g_i^+(x^k)$, $0 < A, < 1$, $i = 1, \ldots, m$, $\sum_{i=1}^{m} \lambda_i = 1$, and $\epsilon_1 \leqslant \alpha_k \leqslant 2 - \epsilon_2, \forall k \leqslant 0$, for some $\epsilon_1, \epsilon_2 > 0$.

Theorem. *If for all $i = 1, \ldots, m$, $g_i(x)$ are continuous and convex functions on $\mathbb{R}^n$, $Q \subseteq \mathbb{R}^n$ is a closed and convex set, $S \neq \emptyset$, and for some $\hat{x} \in S$ there is a constant $K \equiv K(\hat{x})$ such that $\| t \| \leqslant K$ for all $t \in \partial g_i^+(x)$ for all $i = 1, \ldots, m$ and for all $x \in Q$ for which $\| x - \hat{x} \| \leqslant \| x^0 - \hat{x} \|$ (this assumption will be referred to as 'the uniform boundedness of the subgradients'), then the sequence $\{x^k\}$, produced by the PSPA method, converges to a solution of the convex feasibility problem, i.e., $x^k \to x^* \in S$.*

Proof. (1) First, it is shown that $\| x^k - x \| \leqslant \| x^{k+1} - x \|$ for all $x \in S$. For any given $x \in S$ and denoting

$$\gamma_k = \alpha_k \beta_k/\| v^k \|^2,$$

$$\| x^{k+1} - x \|^2 = \left\| P_Q(\tilde{x}^{k+1}) - x \right\|^2 \leqslant \| \tilde{x}^{k+1} - x \|^2 = \left\| x^k - \gamma_k v^k - x \right\|^2$$

because x = $P_Q(x)$ (since x ∈ Q) and the projection operator is non-expansive. Now,

$$\| x^{k+1} - x \|^2 \leqslant \| x^k - x \|^2 + \gamma_k^2 \| v^k \|^2 - 2\gamma_k \langle v^k, x^k - x \rangle$$

and,

$$\langle v^k,\ x^k - x \rangle = \left\langle \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)}{\| t_i^k \|^2} t_i^k,\ x^k - x \right\rangle$$

$$= \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)}{\| t_i^k \|^2} \langle t_i^k,\ x^k - x \rangle .$$

From the subgradient inequality and taking into account the fact that $g_i^+(x) = 0$ for x ∈ S, it is obtained

$$\| x^{k+1} - x \|^2 \leqslant \| x^k - x \|^2 + \gamma_k^2 \| v^k \|^2 - 2\gamma_k \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)^2}{\| t_i^k \|^2}$$

$$= \| x^k - x \|^2 + \frac{\alpha_k^2 \beta_k^2}{\| v^k \|^2} - 2\frac{\alpha_k \beta_k^2}{\| v^k \|^2} = \| x^k - x \|^2 + (\alpha_k^2 - 2\alpha_k) \frac{\beta_k^2}{\| v^k \|^2} .$$

From the fact that $\alpha_k \in [\epsilon_1, 2 - \epsilon_2]$, for all $k \geqslant 0$ we get

$$\| x^{k+1} - x \|^2 \leqslant \| x^k - x \|^2 - \epsilon_1 \epsilon_2 \beta_k^2 / \| v^k \|^2 . \tag{1}$$

Then,

$$\| x^{k+1} - x \| \leqslant \| x^k - x \| \quad \text{for all} \quad x \in S.$$

(2) Now, it is shown that lim $g_i^+(x^k) = 0$ for all $i = 1, \ldots, m$.

For x ∈ S the sequence $\{ \| x^k - x \| \}$ is monotonically decreasing, therefore

$$\lim_{k \to \infty} \| x^k - x \| = d, \quad \text{say}.$$

This implies at once, via (1), that

$$\lim_{k \to \infty} \beta_k / \| v^k \| = 0 \tag{2}$$

Now,

$$\| v^k \| = \left\| \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)}{\| t_i^k \|^2} t_i^k \right\| \leqslant \sum_{i=1}^{m} \lambda_i \frac{g_i^+(x^k)}{\| t_i^k \|} .$$

Denoting $\omega_i^k = g_i^+(x^k) / \| t_i^k \| \geqslant 0$, we get

$$\frac{\beta_k}{\| v^k \|} \geqslant \sum_{i=1}^{m} \lambda_i (\omega_i^k)^2 / \sum_{i=1}^{m} \lambda_i \omega_i^k \geqslant 0.$$

Taking the limits as $k \to \infty$ *we* conclude that

$$\lim_{k \to \infty} \sum_{i=1}^{m} \lambda_i (\omega_i^k)^2 / \sum_{i=1}^{m} \lambda_i \omega_i^k = 0,$$

and it is easy to show that this implies

$$\lim_{k\to\infty} \omega_i^k = \lim_{k\to\infty} g_i^+(x^k)/\|t_i^k\| = 0, \quad i = 1, \ldots, m.$$

Let $\hat{x}$ be the point whose existence is assumed in the assumption of the uniform boundedness of the subgradients. Let

$$S_{\hat{x}} \triangleq \{ x \in Q \mid \|x - \hat{x}\| \leqslant \|x^0 - \hat{x}\| \} \tag{3}$$

then $x^k \in S_{\hat{x}}$ for every $k \geqslant 0$, because $\hat{x} \in S, x^k \in Q$ and by repeated application of (1). Then $\|t_i^k\| \leqslant K$, from which

$$\lim_{k\to\infty} g_i^+(x^k) = 0 \text{ for all } i = 1, \ldots, m. \tag{4}$$

follows.

(3) Finally, it is proved that $\lim_{k\to\infty} x^k = x^* \in S$. Since $\{x^k\} \subseteq S_{\hat{x}}$, which is a compact set, any convergent subsequence of $\{x^k\}$ must satisfy, because Q is closed,

$$\lim_{m\to\infty} x^{k_m} = x^* \in Q.$$

For every $i = 1, \ldots, m$ (4) holds for this subsequence, so, by continuity of $g_i^+, g_i^+(x^*) = 0$, which proves that $x^* \in S$.

In (2) it was showed that

$$\lim_{k\to\infty} \|x^k - x^*\| = d,$$

but now it is known that

$$\lim_{m\to\infty} \|x^{k_m} - x^*\| = 0.$$

Thus, $\lim_{k\to\infty} \|x^k - x^*\| = 0$, and the proof is complete. □

6. **Numerical experiments**

The methods CSP, PSP and PSPA were tested for a number of classical functions (see Moré et all [11]) and were performed in computer PDP-10, in Fortran source code. The algorithms stop when $g_i^+(x) \leqslant \epsilon$ for all $i = 1, \ldots, m$. For ϵ it was chosed the value 10^{-4}.

The test functions were the following:

(1) *Freudenstein and Roth function (non convex).* $n = 2, m = 2,$

$$g_1(x) = -13 + x_1 + ((5 - x_2)x_2 - 2)x_2,$$

$$g_2(x) = -29 + x_1 + ((x_2 + 1)x_2 - 14)x_2.$$

Case I: $x^0 = (10, 4)$,
Case II: $x^0 = (100, 40)$,
Case III: $x^0 = (1000, 400)$.

The results are presented in Table 1.

Table 1

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA {2}
I	0.5	*300, **	*300, **	*300, **	*300, **	*300, **
	1.0	*300, **	*300, **	*300, **	*300, **	*300, **
	1.5	*235,470*	300, *	*300, **	*300, **	*300, **
II	0.5	300, *	300, *	300, *	141,200	179,254
	1.0	300, *	300, *	300, *	26, 33	300, *
	1.5	300, *	300, *	300, *	300, *	28, 36
III	0.5	300, *	300, *	300, *	300, *	300, *
	1.0	300, *	300, *	300, *	233,303	300, *
	1.5	45, 58	300, *	300, *	14, 16	300, *

(2) Modified Jennrich and Sampson function. $n = 2$, $m = 10$,

$$g_i(x) = \exp(ix_1) + \exp(ix_2) - 2i - 2.$$

Case I: $x^0 = (3, 4)$,
Case II: $x^0 = (30, 40)$,
Case III: $x^0 = (300, 400)$.
The results are presented in Table 2.

(3) *Powell singular function.* $n = 4$, $m = 4$,

$$g_1(x) = x_1 + 10x_2, \qquad g_2(x) = \sqrt{5}\,(x_3 - x_4),$$

$$g_3(x) = (x_2 - 2x_3)^2, \qquad g_4(x) = \sqrt{10}\,(x_1 - x_4)^2.$$

Case I: $x^0 = (3, -1, 0, 1)$,
Case II: $x^0 = (30, -10, 0, 10)$,
Case III: $x^0 = (300, -100, 0, 100)$.
The results are presented in Table 3.

Table 2

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA {2}
I	0.5	1, 8	25,105	200, *	9, 84	9, 83*
	1.0	1, 5	8, 52	31, 75	5, 47	5, 47
	1.5	1, 3	5, 46	5, 45	5, 45	5, 44
II	0.5	5, 48	34,196	19,183	200, *	200, *
	1.0	5, 42	8, 78	7, 69	67,660	67,669
	1.5	4, 37	5, 43	5, 46	45,448	44,440
III	0.5	41,408	200, *	91,897	200, *	200, *
	1.0	40,400	44,438	44,437	200, *	200, *
	1.5	40,396	29,283	29,286	200, *	200, *

Table 3

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA (2)
I	0.5	19, 32	15,137	135,241	22, 40	24, 43
	1.0	9, 17	36, 66	200, *	9, 17	20, 34
	1.5	6, 11	23, 42	33, 60	5, 9	7, 11
II	0.5	30, 57	109,188	185,345	30, 53	32, 59
	1.0	14, 28	53, 92	200, *	14, 25	22. 39
	1.5	8, 16	36, 62	95,181	7, 14	9, 15
III	0.5	40, 80	157, 309	200, *	37, 74	40, 75
	1.0	19, 38	78,152	200, *	16, 32	32, 60
	1.5	11, 21	51, 99	200, *	9,17	21, 38

(4) *Modified Wood function. n = 4, m = 6,*

$$g_1(x) = 10\left(x_1^2 - x_2\right), \qquad g_2(x) = x_1 - 1, \qquad g_3(x) = \sqrt{90}\left(x_3^2 - x_4\right),$$

$$g_4(x) = x_3 - 1, \qquad g_5(x) = \sqrt{10}\,(2 - x_2 - x_4), \qquad g_6(x) = \frac{1}{\sqrt{10}}(x_4 - x_2).$$

Case I: x^0 = *(3, -1, 3,* -1),
Case II: x^0 = (30, $-$ 10, 30, -10),
Case III: x^0 = (300, -- 100, 300, $-$ 100).
The results are presented in Table 4.

(5) Modified Extended Rosenbrock function. n = 10, *m* = 10,

$$g_{;\_,}(x) = 10\left(x_{2i-1}^2 - x_{2i}\right),$$

$$g_{2i}(x) = 1 - x_{2i-1},\ i = 1, \ldots, 5$$

Case I: $x^0 = (\xi_j)$ where ξ_{2i-1} = -1.2 and ξ_{2i} = 1, i = 1,..., 5,
Case II: $x^0 = (10\xi_j)$,
Case III: $x^0 = (100\xi_j)$.
The results are presented in Table 5.

Table 4

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA (1)	PSPA {2}
I	0.5	17, 31	130,234	200, *	19, 39	21, 42
	1.0	1, 5	62,108	200, *	3, 9	5, 16
	1.5	1, 3	40, 68	27, 95	2, 6	2, 6
II	0.5	19, 43	148,328	200, *	23, 59	22, 57
	1.0	1, 5	71,151	153,257	4, 14	6, 19
	1.5	1, 5	45, 93	200, *	3, 10	10, 35
III	0.5	23, 55	176,392	200, *	27, 67	48,141
	1.0	1, 5	84,182	77,233	6, 24	6, 22
	1.5	3, 9	54,114	72, 170	4, 12	4, 11

Table 5

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA {2}
I	0.5	15, 80	100, *	100, *	15, 80	20, 63
	1.0	1, 10	95,480	100, *	2, 15	5, 24
	1.5	2, 20	62,315	100, *	3, 20	5, 25
II	0.5	17, 95	100, *	100, *	18,105	21, 58
	1.0	1, 10	100, *	100, *	2, 15	4, 20
	1.5	2, 15	71,385	100, *	2, 15	4, 20
III	0.5	20,115	100, *	100, *	22,135	26, 78
	1.0	1, 10	100, *	100, *	3, 25	7, 29
	1.5	2, 15	82,465	100, *	3, 20	4, 19

(6) *Modified Broyden Tridiagonal function. n = 10, m = 10,*

$$g,(x) = (2x_i - 3)x_i + x_{i-1} + 2x_{i+1} - 1 \quad \text{with } x_0 = x_{n+1} = 0.$$

Case I: $x^0 = (-1, \ldots, -1)$,
Case II: $x^0 = (-10, \ldots, -10)$,
Case III: $x^0 = (-100, \ldots, -100)$.
The results are presented in Table 6.

(7) *Penalty function I. n = 10, m = 11,*

$$g_i(x) = \sqrt{a}\,(x_i - 1), \quad 1 \leqslant i \leqslant n,$$

$$\text{g,+,(x)} = \left(\sum_{i=1}^{m} x_j^2 \right) - \tfrac{1}{4} \quad \text{with } a = 10^{-5}.$$

Case I: $x^0 = (1, 2, \ldots, 9, 10)$,
Case II: $x^0 = (10, 20, \ldots, 90, 100)$,
Case III: $x^0 = (100, 200, \ldots, 900, 1000)$.
The results are presented in Table 7.

Table 6

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA {2}
I	0.5	64,624	100, *	100, *	37,310	46,289
	1.0	21, 200	100, *	100, *	35,196	22, 89
	1.5	10, 47	100, *	100, *	7, 33	11, 4
II	0.5	79,766	100, *	100, *	39,346	56,192
	1.0	26,253	100, *	100, *	34,190	32, 96
	1.5	11, 63	100, *	100, *	8, 41	9, 38
III	0.5	88, 849	100, *	100, *	48,420	84, 243
	1.0	30,285	100, *	100, *	27,164	81, 293
	1.5	13, 81	100, *	100, *	9, 57	29,137

Table 7

Case	α_k	CPS	PSP {1}	PSP (2)	PSPA (1)	PSPA (2)
I	0.5	18, 90	198,990	200, *	18, 90	34,136
	1.0	9, 45	99,945	200, *	9, 45	20, 68
	1.5	6, 33	66,333	200, *	6, 33	11, 48
II	0.5	198, 1080	200, *	200, *	198, 1080	200, *
	1.0	99,540	200, *	200, *	99,540	173,722
	1.5	66,363	200, *	200, *	66,363	141,573
III	0.5	200, *	200, *	200, *	200, *	200, *
	1.0	200, *	200, *	200, *	200, *	200, *
	1.5	200, *	200, *	200, *	200, *	200, *

(8) Variable dimensioned function, n = 10, m = 12,

$$g_i(x) = x_i - 1, \quad 1 \leqslant i \leqslant n,$$

$$g_{n+1}(x) = \sum_{j=1}^{n} j(x_j - 1), \qquad g_{n+2}(x) = \Big(\sum_{j=1}^{n} j(x_j - 1)\Big)^2.$$

Case I: $x^0 = (0.9, 0.8, \ldots, 0.1, 0)$,
Case II: $x^0 = (9, 8, \ldots, 1, 0)$,
Case III: $x^0 = (90, 80, \ldots, 10, 0)$.
The results are presented in Table 8.

The pair k, p means that the method converged in k iterations and p projections were computed; the pair k, * means that the method have not converged in k iterations. The number {1} indicates that the coefficients in the convex combination are equal and { 2) indicates that they are differents (random generation).

Remark. For the CPS method one iteration corresponds to a complete cycle, i.e., m approximated projections,

Table 8

Case	α_k	CPS	PSP {1}	PSP {2}	PSPA {1}	PSPA {2}
I	0.5	12, 77	188, 1117	200, *	20,161	47,216
	1.0	1, 10	91,533	200, *	3, 35	26,109
	1.5	3, 13	58,339	200, *	4, 15	33,152
II	0.5	200, *	200, *	200, *	200, *	200, *
	1.0	200, *	200, *	200, *	200, *	200, *
	1.5	200, *	200, *	200, *	200, *	200, *
III	0.5	200, *	200, *	200, *	200, *	200, *
	1.0	200, *	200, *	200, *	200, *	200, *
	1.5	200, *	200, *	200, *	200, *	200, *

7. Conclusions

The methods introduced in this work, PSP and PSPA, are new methods to solve iteratively the convex feasibility problem.

It may be conclude that:

(1) The PSP method is less efficient than the CSP method. It is because that the speed of convergence of the PSP method is very low when the number of violated restrictions ($g_i^+(x^k) > 0$) is small.

(2) The convex combination with equal coefficients is more efficient in the PSP method, while in the accelerated version a defined conclusion does not exist.

(3) The generalization of the scheme of acceleration really improves the speed of convergence of the PSP method.

(4) The methods are more efficient for values of $\alpha_k > 1$, since we are working with inequalities.

Comparing the number of iterations in the PSPA method with the number of projections in the CSP method, it was noted that the first is more efficient when parallel computation is possible, since the calculus of the terms of β_k and v^k may be evaluated at the same time, i.e., in parallel.

Other advantages of the methods are:

(1) No minimization problems need to be solved at each step, due to the use of subgradient projections.

(2) The computational implementation is easy, and its storage requirements are very low.

The numerical experiences show that the PSPA method is a potentially powerful tool under the parallel computation context.

Further research is necessary in order to specify an appropriate way for choosing the relaxation parameter α_k. The main advantage of its introducing is to show that the determination of the acceleration step does not need to be very accurate. A direct consequence of this fact is the possible consideration of alternative formula for the acceleration step.

Acknowledgement

The author wishes to express his thanks to Dr. José Mário Martinez-Supervisor of this work-, and to FAPESP for the encouragement and support given him throughout this work.

References

[1] Y. Censor, Row-action methods for huge and sparse systems and their applications, *SIAM Reu. 23 (1981) 444-466.*

[2] *Y.* Censor, Finite series expansion reconstruction Methods, *Proc. IEEE 71 (1983)* 409–419.

[3] *Y.* Censor and T. Elfving, New methods for linear inequalities, *Lin. Alg. Appl. 42 (1982) 199-211.*

[4] *Y.* Censor and A. Lent, A cyclic subgradient projections method for the convex feasibility problems, Technical Report, University of Haifa, Israel, 1980.

[5] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, *La Ricerca Scientifica. Roma. Ser. II, Anno IX.* 1 (1938) 326-333.

[6] A.R. de Pierro, Metodos de projeção para a resolução de sistemas gerais de equações algébricas lineares, Tese de Doutoramento, IM-UFRJ, 1981.
[7] A.R. de Pierro and A.N. Iusem, A parallel projection method of finding a common point of a family of convex sets, Informes de Matemática, Serie B-021/84, IMPA, Rio de Janeiro, 1984.
[8] L.G. Gubin, B.T. Polyak and E.V. Raik, The method of projections for finding the common points of convex sets. ***USSR Comp. Math. Phys. 7 (1967) l-24.***
[9] ***S.*** Kaczmarz, Angenaherte auflösung von systemen linearer gleichungen, *Bull. Acad. Polon.* ***Sci. Lett. A 35 (1937) 355-357.***
[10] J.M. Martinez, Solving systems of nonlinear equations by means of an accelerated successive orthogonal projections method, ***J. Comput. Appl. Math 16 (1986) 169-179.***
[11] J.J. More, B.S. Garbow and K.E. Hillstrom, Testing unconstrained Optimization Software, ***ACM Trans. Math. Software 7 (1981) 17-41.***
[12] H. Mukai, Parallel algorithms for solving systems of nonlinear equations, ***Comput. Math. Appl. 7 (1981)*** 235–250.
[13] L.T. Santos, Mttodos de projeção do subgradiente para o problema de factibilidade convexa, Tese de Mestrado, IMECC-UNICAMP, 1985.
[14] R.L. Wainwright, Four dimensional x-projection method (with acceleration techniques) for solving systems of linear equations, ***Comput. Math. Appl. 8 (1982) 329-337.***